Refined Ranking Relations for Selection of Solutions in Multi Objective Metaheuristics

Ruby L. V. Moritz, Enrico Reich, Maik Schwarz, Matthias Bernt, and Martin Middendorf

Dept. of Computer Science, University of Leipzig, Leipzig, Germany

Abstract

Two methods for ranking of solutions of multi objective optimization problems are proposed in this paper. The methods can be used, e.g. by metaheuristics to select good solutions from a set of non dominated solutions. They are suitable for population based metaheuristics to limit the size of the population. It is shown theoretically that the ranking methods possess some interesting properties for such applications. In particular, it is shown that both methods form a total preorder and are both refinements of the Pareto dominance relation. An experimental investigation for a multi objective flow shop problem shows that the use of the new ranking methods in a Population-based Ant Colony Optimization algorithm and in a genetic algorithm leads to good results when compared to other methods.

Keywords: Metaheuristics, Multi objective optimization, Ranking Relations, Ant Colony Optimization, Genetic algorithms, Flow Shop Scheduling Problem

1. Introduction

Optimization problems with several objectives are common in industrial, engineering, or scientific contexts. Abstractly formulated, a multi objective optimization problem asks for solutions from a solution space \( X \) (also called search space) that are optimal with respect to \( d > 1 \) objectives. Typically, the objectives are conflicting and it is only possible to optimize a (small) subset of the objectives simultaneously. This is reflected in the concept of the Pareto dominance relation. A solution dominates another if it is better in at least one objective and not worse in all other objectives. Based on this relation a solution is called Pareto optimal if it is not dominated by any other solution from \( X \). The goal in multi objective optimization is to find the set of Pareto optimal solutions, called Pareto set, or at least a subset of it. Unfortunately, many discrete multi objective optimization problems are NP-hard, i.e. it is not possible to solve them in polynomial time (if \( P \neq NP \)). For continuous multi objective optimization problems the optimization function is often a black box or algebraically too complicated, i.e. it is impossible to solve them.
analytically. In these cases, usually the goal is to find a set of non dominated solutions that are close to the Pareto optimal solutions. Another common selective feature is the diversity of the chosen set of solutions.

To explore the solution space the use of algorithms maintaining a population of solutions seems beneficial. Therefore, multi objective variants of various population-based metaheuristics have been developed in recent years. Among the earliest multi objective variants of genetic algorithms is, e.g. the Vector Evaluation Genetic Algorithm (VEGA) from Schaffer (1984) (see overviews in Coello Coello (2009), Coello Coello et al. (2005), and Fonseca and Fleming (1995)). Also the ant colony optimization metaheuristic (ACO) was extended to solve multi objective optimization problems, e.g. by Iredi et al. (2001) and Doerner et al. (2001) (for an overview see Angus and Woodward (2009) or Leguizamón and Coello Coello (2011)). For problems on continuous solution spaces extended particle swarm approaches (PSO) have been proposed, e.g. by Coello Coello and Salazar Lechuga (2002) and Hu and Eberhart (2002) (for an overview see Reyes-Sierra and Coello Coello (2006)).

The Pareto dominance relation is often used in population-based metaheuristics to rank the elements of a set of solutions. But for a set of non dominated solutions this relation is insufficient to guide heuristics into favorable regions of the search space. This is particularly relevant, when the number of objectives is large since the number of non dominated solutions increases along with the number of objectives. This effect has been called a “curse of dimensionality” by Kukkonen and Lampinen (2007). Therefore, several methods to compare and rank elements of sets of (non dominated) solutions have been discussed in the literature. Some of these ranking methods are used for post-Pareto optimality, i.e. to select solutions from the final set of non dominated solutions computed by some metaheuristic.

Aside from ranking the found solutions, there are other methods to prune a set of (non dominated) solutions. One possibility is to apply clustering methods to the set of solutions in order to select a representative solution from each cluster. One potential aim of using clustering methods is to obtain a large diversity in the set of selected solutions (Kukkonen and Deb, 2006).

In this paper we are interested in using ranking methods for metaheuristics by applying them in each iteration to select only a few solutions from the set of already found (non dominated) solutions. Hence, the ranking methods are used to prune a potentially large set of solutions to a small subset of good solutions. This small set forms the population which is then used in the next iteration to create new and hopefully better solutions. Maintaining a small population has advantages with respect to the memory and time requirements of an algorithm. This is particularly important in restrictive computational environments, e.g. when solutions need to be delivered in real time or the algorithm runs on specific hardware, e.g. as part of an embedded system. Another important reason for the use of small populations are objective functions that are expensive to evaluate. Clearly, the choice of which solutions are to be kept in the population is particularly important for small population sizes.

The simplest kind of ranking schemes are aggregation methods. They use, for instance, an aggregation function calculating a weighted sum of the objectives (Jakob et al., 1992) or the distance to a target vector of objective values (Wienke et al., 1992). Furthermore, there are also objective based ranking methods. One such method is to use given priorities for a lexicographic sorting of the objectives or performing a separate optimization of the single objectives in an order that is compliant with the priorities (Cvetkovic and Parmee, 2002). Weights, target vectors, and priorities should be given by the user. But for many applications this might be difficult or impossible for the user because, for instance, no reasonable weights are known.

In contrast to the above ranking methods that need different kinds of user input, several
methods have been developed based on the notion of dominance. To rank a solution $a$ within a set of solutions $X$ one could use the number of solutions in $X$ that are dominated by $a$ (dominance rank) or the number of solutions in $X$ dominating $a$ (dominance count) (see also Fonseca and Fleming (1993) and Zitzler and Thiele (1999)). An alternative is to use the number of times the current set of non dominated solutions has to be removed from the remaining set of solutions until the solution itself becomes non dominated. This measure has been called dominance depth by Srinivas and Deb (1994). Additionally to the comparison of single objective vectors also sets of non dominated solutions can be compared. This is commonly done in the indicator function framework, which assigns a real value to a set or sets of non dominated solutions. One example is the binary hypervolume indicator that is defined as the hypervolume of the objective space (i.e. the space of vectors of potential values of the different objectives) dominated by one set of solutions but not by another (we refer the interested reader to Zitzler et al. (2003)).

Some ranking methods consider how much two solutions differ with respect to the different objectives. Garza-Fabre et al. (2009) have proposed three such methods. An example is the Global Detriment method which computes for each solution the sum of the differences to other solutions over all those objectives where the solution is worse than the corresponding other solution. A potential disadvantage of these methods is that they typically need some normalization between the different objectives. Therefore, these methods are not considered further in this paper. However, we use one representative of these methods as a comparison method. This method uses the same measure as the Global Detriment method, but only for pairwise comparison, and random weights for the normalization (Details see Section 3.2).

By regarding only the information whether two objective values differ and thereby disregarding the magnitude of the difference of the objective values the decision process can be greatly simplified. The number of objectives for which one solution is better or worse than the other can be helpful for a decision between alternative solutions. A similar reasoning simplifies everyday decisions when comparing different choices by the number of advantages and disadvantages. This basic idea was formally captured by the relation favour that prefers a solution over another if it wins, i.e. is better, in more objectives than the other (Drechsler et al., 2001). Several extensions or similar relations have been proposed in the literature. They modify the decision if an objective is counted as won or lost (Laumanns et al., 2002; Sülflow et al., 2007) or how many won objectives are required for preference (Farina and Amato, 2004; Zou et al., 2008). Approaches based on the number of won objectives can also be used to rate a solution with respect to a set of solutions e.g., (Bentley and Wakefield, 1998; Maneeratana et al., 2006; Mostaghim and Schmeck, 2008). A detailed description of these methods is given in Section 3.

In this paper we propose two new ranking relations which are based on the relation favour (Section 4). We prove that both relations are refinements of the Pareto dominance relation and are total preorders. The ranking relations can be used in multi objective population based metaheuristics for selecting the solutions that are included into the population and thus guide the search for better solutions. We compare the different ranking schemes when used in a Population-based Ant Colony Optimization algorithm (P-ACO) and in a genetic algorithm (GA). As test problem a multi objective flow shop scheduling problem is used. The results show that the new ranking schemes are advantageous to select the solutions for the population of the P-ACO and the GA.

The paper starts with the introduction of basic definitions in the next Section 2. A detailed review of related ranking relations from the literature and a few corresponding theoretical results are presented in Section 3. The new ranking relations are introduced in Section 4. The metaheuristics P-ACO and GA which are used for the experiments are described in Section 5. The test
problem and the experiments are described in Section 6. The experimental results are presented in Section 7. A short conclusion is given in Section 8. Note, that this paper is an extension of Moritz et al. (2013).

2. Basic Definitions

Consider a multi objective optimization problem with a set of solutions $X$ and a vector of objective functions $\mathbf{f}(a) = (f_1(a), \ldots, f_d(a))$, where $a \in X$ and $f_i : X \mapsto \mathbb{R}$. The solutions in $X$ can, for example, be vectors of real values in case of continuous optimization problems or vectors of elements from a finite set in case of combinatorial optimization problems. The aim is to find solutions from $X$ that minimize the objectives, i.e.

$$\min_{a \in X} \mathbf{f}(a) = \min_{a \in X} (f_1(a), \ldots, f_d(a)).$$  \hspace{1cm} (1)

Note, that by using $-f_i(a)$ maximization is also possible.

In order to find a minimum in a two or higher dimensional space the Pareto dominance relation ($\prec$) is used. Let $a, b \in X$, then

$$a \prec b \iff \forall i \in [1 : d] : f_i(a) \leq f_i(b) \wedge f_i(a) \neq f_i(b).$$  \hspace{1cm} (2)

Solution $a$ dominates $b$ if $a < b$. Note that, if $a < b$ then there is at least one $i \in [1 : d]$ with $f_i(a) < f_i(b)$. Two solutions $a, b \in X$ are called incomparable if $a \not< b$ or indifferent in case of $\mathbf{f}(a) = \mathbf{f}(b)$. A solution $a \in X$ is called Pareto optimal if $\not\exists b \in X : a < b$. A solution $a \in X$ is called non dominated solution with respect to a subset $X' \subseteq X$ if $\not\exists b \in X' : a < b$. The set of all Pareto optimal solutions from $X$ is called the Pareto set and the corresponding set of objective vectors in $\mathbb{R}^d$ is called the Pareto front of $X$.

The different properties of relations are summarized in Table 1. Let $R$ and $S$ be two relations on a set $X$. Then $S$ is a refinement of $R$ if

$$\forall a, b \in X : (a, b) \in R \wedge (b, a) \notin R \Rightarrow (a, b) \in S \wedge (b, a) \notin S.$$  

In terms of the domination relation a refinement is useful as it preserves the original order and elements that are incomparable with the domination relation could be comparable with respect to the refinement.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>reflexive</td>
<td>$\forall a \in X : (a, a) \in R$</td>
</tr>
<tr>
<td>irreflexive</td>
<td>$\forall a \in X : (a, a) \notin R$</td>
</tr>
<tr>
<td>transitive</td>
<td>$\forall a, b, c \in X : (a, b), (b, c) \in R \Rightarrow (a, c) \in R$</td>
</tr>
<tr>
<td>symmetric</td>
<td>$\forall a, b \in X : (a, b) \in R \Rightarrow (b, a) \in R$</td>
</tr>
<tr>
<td>asymmetric</td>
<td>$\forall a, b \in X : (a, b) \in R \Rightarrow (b, a) \notin R$</td>
</tr>
<tr>
<td>antisymmetric</td>
<td>$\forall a, b \in X : (a, b), (b, a) \in R \Rightarrow a = b$</td>
</tr>
<tr>
<td>total</td>
<td>$\forall a, b \in X : (a, b) \in R \vee (b, a) \in R$</td>
</tr>
<tr>
<td>preorder on $X$</td>
<td>$R$ reflexive and transitive</td>
</tr>
<tr>
<td>total preorder on $X$</td>
<td>$R$ is a preorder on $X$ and total</td>
</tr>
<tr>
<td>partial order on $X$</td>
<td>$R$ is reflexive, transitive, and antisymmetric</td>
</tr>
<tr>
<td>total order on $X$</td>
<td>$R$ is a partial order on $X$ and total</td>
</tr>
</tbody>
</table>
Given an irreflexive and asymmetric relation \( R \) over a set \( W \), two vectors \( \vec{u}, \vec{v} \in W^d \) are compared by:

- \( \mathcal{B}^R(\vec{u}, \vec{v}) = |\{ i : (\vec{u}_i, \vec{v}_i) \in R, i \in [1 : d] \}| \)
- \( \mathcal{E}^R(\vec{u}, \vec{v}) = |\{ i : (\vec{u}_i, \vec{v}_i) \notin R, (\vec{v}_i, \vec{u}_i) \notin R, i \in [1 : d] \}| \)
- \( \mathcal{W}^R(\vec{u}, \vec{v}) = |\{ i : (\vec{v}_i, \vec{u}_i) \in R, i \in [1 : d] \}| \)

Note that due to the asymmetry and irreflexivity each element \( i \in [1 : d] \) is included in exactly one of the three sets used to compute \( \mathcal{B}^R(\vec{u}, \vec{v}), \mathcal{E}^R(\vec{u}, \vec{v}) \), and \( \mathcal{W}^R(\vec{u}, \vec{v}) \). Thus, the sum of the sizes of the three sets is \( d \), i.e. \( \mathcal{B}^R(\vec{u}, \vec{v}) + \mathcal{E}^R(\vec{u}, \vec{v}) + \mathcal{W}^R(\vec{u}, \vec{v}) = d \). This notion is used to compare elements of the solution space of multi objective optimization problems with respect to their values in the objective space \( \mathbb{R}^d \) using the \( < \) relation. In the following we write \( \mathcal{B}^R(a, b) \) instead of \( \mathcal{B}^R(\vec{f}(a), \vec{f}(b)) \) whenever we compare two solutions \( a, b \in X \). We handle \( \mathcal{E}^R(a, b) \) and \( \mathcal{W}^R(a, b) \) analogously. In that case \( \mathcal{B}^R(a, b), \mathcal{E}^R(a, b), \) and \( \mathcal{W}^R(a, b) \), respectively, counts the number of objectives where a solution \( a \) is better (respectively equal, worse) than a solution \( b \) (Drechsler et al., 2001; Farina and Amato, 2004; Sülflow et al., 2007; Zou et al., 2008).

For a solution \( u \in X \), a set \( X' \subseteq X \) of solutions and an irreflexive and asymmetric relation \( R \subseteq X \times X \) a similar notion is introduced:

- \( \mathcal{B}^R(u, X') = |\{ x : (u, x) \in R, x \in X' \}| \)
- \( \mathcal{E}^R(u, X') = |\{ x : (u, x) \notin R, (x, u) \notin R, x \in X' \}| \)
- \( \mathcal{W}^R(u, X') = |\{ x : (x, u) \in R, x \in X' \}| \)

Note, that \( \mathcal{B}^R(u, v) \) and \( \mathcal{B}^R(u, |v|) \) have a different meaning. For \( X'', X' \subseteq X \) define \( \mathcal{B}^R(X'', X') = \sum_{x \in X''} \mathcal{B}^R(u, X') \). Analogously \( \mathcal{E}^R(X'', X') \) and \( \mathcal{W}^R(X'', X') \) are defined.

### 3. Related Ranking Relations

In this section we review different approaches that have been proposed in the literature to compare non dominated solutions based on certain relations between their objective values.

#### 3.1. Relation favour

The relation favour (Drechsler et al., 2001) compares two solutions with respect to the number of objectives in which one solution is better than the other solution. This is captured by the following formal definition.

\[
a \prec_f b \iff \mathcal{B}^f(a, b) > \mathcal{B}^f(b, a)
\]

(3)

Obviously, \( \prec_f \) is an irreflexive and asymmetric relation. But relation favour is not transitive (Drechsler et al., 2001). Since ties in the number of objectives where one solution is better than the other and vice versa are possible relation favour is not total. Moreover, for \( d \leq 2 \), i.e. for two or one objective, the favour relation and the Pareto dominance relation are equivalent (Drechsler et al., 2001). Relation favour is a refinement of the Pareto dominance relation. Since this has not been noticed in the literature so far (to the best of our knowledge) we provide proof in the following.

**Theorem 3.1.** The favour relation is a refinement of the Pareto dominance relation on \( X \), i.e. for \( a, b \in X \) : \( a \prec_f b \rightarrow a \prec a \land b \not\succ_f a \).

**Proof.** It suffices to show that \( a \prec b \) implies \( a \prec_f b \) as the second part of premise and conclusion are a direct consequence of the asymmetry of \( \prec \) and \( \prec_f \). By definition of the Pareto dominance relation it holds that \( \exists i \in [1 : d] : f_i(b) < f_i(a) \), i.e. \( \mathcal{B}^f(b, a) = 0 \), and \( \exists i \in [1 : d] : f_i(a) < f_i(b) \), i.e. \( \mathcal{B}^f(a, b) > 0 \). This implies \( a \prec_f b \). \( \square \)
In addition, we show some properties on the combination of the Pareto dominance relation and the favour relation used in the next section.

**Proposition 3.1.** Let \( a, b, c \in X \) with \( a < b \). Then: (i) \( b <_{f} c \Rightarrow a <_{f} c \), (ii) \( c \not<_{f} b \Rightarrow c \not<_{f} a \), (iii) \( c <_{a} a \Rightarrow c <_{f} b \) (iv) \( a \not<_{f} c \Rightarrow b \not<_{f} c \).

**Proof.** Because \( a < b \), with \( a, b \in X \), for any \( c \in X \) it holds that
(1) \( B'(b, c) \leq B'(a, c) \) because \( \forall k \in [i : f_i(b) < f_i(c)] : f_i(a) \leq f_i(b) \), and
(2) \( B'(c, b) \geq B'(c, a) \), because \( \forall k \in [i : f_i(c) < f_i(a)] : f_i(a) \leq f_i(b) \).

These inequalities together with the corresponding premise of the implication gives the result.
(i) Due to \( b <_{f} c \) we have \( B'(b, c) > B'(c, b) \) yielding: \( B'(a, c) \geq B'(b, c) > B'(c, b) \geq B'(c, a) \).
By definition (Equation (3)) this is equivalent to \( a <_{f} c \). (ii) From \( c \not<_{f} b \) we know \( B'(c, b) \leq B'(b, c) \) and therefore \( B'(c, a) \leq B'(c, b) \leq B'(b, c) \leq B'(a, c) \), i.e. \( c \not<_{f} a \). (iii) Similarly \( c <_{f} a \) gives \( B'(c, b) \geq B'(c, a) \geq B'(b, c) \geq B'(c, a) \), i.e. \( c <_{f} b \). (iv) Finally \( a \not<_{f} c \) gives \( B'(b, c) \leq B'(c, b) \leq B'(c, a) \leq B'(b, c) \), i.e. \( b \not<_{f} c \).

The favour relation on a set \( X \) can be represented by a directed graph \( G = (V, E) \) with node set \( V = X \) and an edge set defined by the favour relation, i.e. \( (b, a) \in E \) iff \( a <_{f} b \). \( G \) is called the *favour graph* of \( X \) and vector \( \vec{f} \) of objective functions. The following lemma shows that every directed graph can be the favour graph of \( V \) and some \( \vec{f} \).

**Lemma 3.1.** For every directed graph \( G = (V, E) \) that is loop-free (i.e. \( \forall a \in V : (a, a) \notin E \) and has at most one edge between two nodes there exists a vector of objective functions \( \vec{f} \) on the set \( X = V \) such that the favour graph of \( V \) and \( \vec{f} \) is isomorphic to \( G \).

**Proof.** Let \( G = (V, E) \) be given with \( V = \{v_1, \ldots, v_n\} \) and \( E = \{e_1, \ldots, e_m\} \). For the construction let \( \vec{f} = \{f_1, \ldots, f_m\} \) be 3\( m \)-dimensional vector of objective functions. For each \( v \in V \) let \( \vec{f}(v) \), with \( h \in [1 \ldots m] \), be the three dimensional subvector of \( \vec{f}(v) \) which defines the positions \( 3h - 2, 3h - 1, \) and \( 3h \) in \( \vec{f}(v) \). Define the objective functions such that for each edge \( e_h = (v_i, v_j) \in E \) it holds that \( \vec{f}(h) = (1, 1, 2), \vec{f}(h) = (2, 2, 1), \) and for all other vectors \( \vec{f}(h) = (1, 3, 1), i \neq j \).

By the construction, it follows that if there exist an edge \( e_h = (v_i, v_j) \in E \) then \( v_i^{(h)} \not<_{f} v_j^{(h)} \) and \( v_i^{(h)} \not<_{f} v_j^{(h)} \) \( v_i^{(h)} \not<_{f} v_j^{(h)} \) \( v_i^{(h)} \not<_{f} v_j^{(h)} \) for \( k \in [1 \ldots m] \). \( k \neq h \). If two nodes \( v_i, v_j \in V \) are not connected by an edge in \( G \) then \( v_i^{(h)} \not<_{f} v_j^{(h)} \) \( v_i^{(h)} \not<_{f} v_j^{(h)} \) \( v_i^{(h)} \not<_{f} v_j^{(h)} \) for all \( h \in [1 \ldots m] \). Altogether it follows for vectors \( v_i, v_j \in V \) that the favour relation \( v_i \not<_{f} v_j \) holds iff \( v_j, v_i \in E \). Hence, the favour graph of \( V \) and \( \vec{f} \) is isomorphic to \( G \).

It should be mentioned that several variations of the favour relation have been proposed. For example the \( \varepsilon \)-preference rates an objective \( f_i, i \in [1 \ldots d] \), of a solution as won only if its value is larger by a constant value \( \varepsilon_i \) than the corresponding value of the competitor solution (Sülfow et al., 2007). This is analogous to a corresponding variation of the dominance relation that was called \( \varepsilon \)-dominance by Laumanns et al. (2002). Other modifications consider an additive or multiplicative tolerance for the number of won objectives, e.g. by demanding \( L + B'(a, b) > B'(a, b) \) or \( k \cdot B'(a, b) > W'(a, b) \) where \( L \) and \( k \) are constants. For values \( L > 0 \) and \( k > 1 \) a stricter form of dominance can be enforced, and for values \( L < 0 \) and \( k < 1 \) weaker forms are enforced. With small modifications this has been implemented in the notion of \( L \)-dominance by Zou et al. (2008) and \( (1 - k) - dominance \) by Farina and Amato (2004), respectively. A solution \( a \) is said to \((1 - k)\)-dominate solution \( b \) (denoted by \( a <_{1-k} b \) iff \( E'(a, b) < d \) and
\[ B(a, b) \geq (d - E(a, b))(k + 1), \] with \( k \in [0 : 1] \). The latter inequality can be transformed to \( k \cdot B(a, b) \geq W(a, b) \). L-dominance for solutions \( a \) and \( b \) is defined by \( B(a, b) = W(a, b) = L > 0 \), i.e. \( B(a, b) - L = W(a, b) \). Zou et al. (2008) demanded in addition that \( a \) is better than \( b \) with respect to some norm, e.g. the sum of the objective values.

3.2. Other Ranking Methods

Instead of comparing two solutions directly it is also interesting to compare them by checking which one performs better with respect to a reference set of solutions. This has been done for instance in the domination rank, domination count, and domination depth based ranking methods. Two other such methods are the winning score by Maneeratana et al. (2006) and the average ranking by Bentley and Wakefield (1998). For a set of solutions \( X \) the winning score of a solution \( a \in A \subset X \) is defined as \( S(a) = \sum_{b \in A} (B(a, b) - W(a, b)) \). The average rank is given by \( \text{rank}(a) = \sum_{i=1}^{d} r_i(a) \) where \( r_i(a) \) is the rank of \( a \) with respect to objective \( i \). Corne and Knowles (2007) have shown that winning score and average rank are equivalent, i.e. they induce the same order on the elements of \( A \).

We use the winning score for a comparison with our new methods in the experimental analysis. The corresponding relation is abbreviated as \( \text{Win} \). In addition, we propose a comparison method — called method \( W \) — that uses a similar measure as the Global Detriment method of Garza-Fabre et al. (2009) (see also Section 1), however only pairwise and it uses weights for normalization. Method \( W \) uses the weighted sum of the objectives for the ranking, i.e. \( a \prec_W b \) if \( \sum_{i=1}^{d} w_i f_i(a) \leq \sum_{i=1}^{d} w_i f_i(b) \).

Note, that the winning score and the relation used in method \( W \) are both refinements of the Pareto dominance relation. For the former this is because \( a \prec b \) implies that the average rank of \( a \) is smaller than the average rank of \( b \). For the latter this is because, \( a \prec b \) implies that the weighted sum of \( a \) is smaller than the weighted sum of \( b \). Furthermore, both relations are total preorders, since \( \leq \) is used as comparison operator in both relations which implies reflexivity, transitivity, and totality. By using \( < \) instead of \( \leq \) reflexivity can be removed.

A good overview and experimental comparison over different ranking methods can be found in Garza-Fabre et al. (2009) and Garza-Fabre et al. (2010).

4. New Refined Ranking Relations

In this section we introduce the following new comparison operations: \( \text{relation WL and relation Points} \). Both relations are based on the relation favour. They are used to compare (non-dominated) solutions.

For a solution \( a \in X \) the comparisons are based on the number of solutions in \( A \subset X \) compared to which \( a \) wins, is incomparable, or loses according to the favour relation, i.e. \( \mathcal{E}(a, A) \), \( \mathcal{E}^*(a, A) \), or \( \mathcal{W}^*(a, A) \) respectively. The following theorem relates Pareto dominance to these numbers.

**Theorem 4.1.** Let \( A \subseteq X \) and \( a, b \in X \). It holds that \( a \prec b \Rightarrow \mathcal{B}^*(a, A) \geq \mathcal{B}^*(b, A) \land \mathcal{W}^*(a, A) \leq \mathcal{W}^*(b, A) \). If \( a, b \in A \) both inequalities are strict.

**Proof.** Assume \( a \prec b \). By Proposition 3.1 for every \( x \) it holds that i) if \( b \prec_f x \) then \( a \prec_f x \) and ii) if \( x \prec_f a \) then \( x \prec_f b \). Hence, the first sentence of the theorem holds. Furthermore, by Theorem 3.1 we have \( a \prec_f b \). Thus, it follows that \( \mathcal{B}^*(a, A) \geq \mathcal{B}^*(b, A) + 1 \) and \( \mathcal{W}^*(a, A) - 1 \leq \mathcal{W}^*(b, A) \) when \( a, b \in A \).

\( \square \)
Note that relation favour may distinguish between non dominated solutions only for multi objective optimization problems with more than two objectives. This is because for two objectives the comparison between two non dominated solutions is always tied. Consequently, the relations introduced in the following have no refining effect in case of two or less objectives as they rely on relation favour.

4.1. Relation WL

Let \( A, b \in X \), and \( \preceq_{\text{WL}} \) be a relation on \( X \) defined as follows.

\[
\begin{align*}
\text{a} & \preceq_{\text{WL}} \text{b} \iff \mathcal{B}(a, A) > \mathcal{B}(b, A) \\
(\mathcal{B}(a, A) = \mathcal{B}(b, A) \land \mathcal{W}(a, A) \leq \mathcal{W}(b, A)) \end{align*}
\]

Relation \( \preceq_{\text{WL}} \) is called WL relation. If the context is clear we may omit \( A \) in the notation. Some favorable properties of relation WL are given in the following.

**Theorem 4.2.** For \( A \subseteq X \) the WL relation \( \preceq_{\text{WL}} \) is a total preorder on \( X \).

**Proof.** We show that the WL relation \( \preceq_{\text{WL}} \) is (i) transitive and (ii) total (which implies it is reflexive). (i) Let \( a, b, c \in X \) with \( a \preceq_{\text{WL}} b \) and \( b \preceq_{\text{WL}} c \) then \( a \preceq_{\text{WL}} c \) follows from the transitivity of \( > \), \( = \), and \( \leq \). (ii) Assume \( a \not\preceq_{\text{WL}} b \), with \( a, b \in X \). Consider the two possible cases: a) \( \mathcal{B}(a, A) < \mathcal{B}(b, A) \) and b) \( \mathcal{B}(a, A) = \mathcal{B}(b, A) \) \( \land \mathcal{W}(a, A) < \mathcal{W}(b, A) \). In each case the definition of the WL relation implies \( b \not\preceq_{\text{WL}} a \).

The theorem ensures the possibility of sorting a set of solutions according to the WL relation with sorting algorithms that are based on pairwise comparisons, e.g. merge sort. Note, that the WL relation is not a partial order since antisymmetry does not hold, i.e. \( a \preceq_{\text{WL}} b \) and \( b \preceq_{\text{WL}} a \) does not imply \( a = b \).

Using \( \mathcal{W}(a, A) < \mathcal{W}(b, A) \) instead of \( \mathcal{W}(a, A) \leq \mathcal{W}(b, A) \) in Equation (4) would result in an irreflexive relation. But this would have the negative side effect that the relation is not total anymore and therefore two different solutions with the same objective values are not comparable.

**Example 4.1.** Let \( X = \{a, b, c\} \) with \( f(a) = (1, 2, 3) \), \( f(b) = (3, 1, 2) \), and \( f(c) = (2, 3, 1) \), i.e. \( a \not< f \), \( c \not< b \), and \( b \not< f \). Then \( \mathcal{B}(a, X) = \mathcal{B}(b, X) = \mathcal{B}(c, X) = 1 \) and \( \mathcal{W}(a, X) = \mathcal{W}(b, X) \) = \( \mathcal{W}(c, X) = 1 \). Hence, \( a \preceq_{\text{WL}} b \preceq_{\text{WL}} c \preceq_{\text{WL}} a \). Since, \( a, b, c \) are pairwise different relation \( \preceq_{\text{WL}} \) is not antisymmetric.

The following proposition is needed to show that the WL relation is a refinement of the Pareto dominance relation.

**Proposition 4.1.** Let \( A \subseteq X \) and \( a, b \in A \). Then \( a \prec b \Rightarrow b \not\preceq_{\text{WL}} a \).

**Proof.** Due to Theorem 4.1 \( a \prec b \) implies \( \mathcal{B}(a, X) > \mathcal{B}(b, A) \). Thus, by definition (see Equation 4) \( b \not\preceq_{\text{WL}} a \) holds. 

A consequence of the previous proposition is the following theorem.

**Theorem 4.3.** The WL relation \( \preceq_{\text{WL}} \) is a refinement of the Pareto dominance relation on \( A \subseteq X \), i.e. for \( a, b \in A \):

\[
a \prec b \land b \not\not\prec a \Rightarrow a \preceq_{\text{WL}} b \land b \not\preceq_{\text{WL}} a.
\]
Proof. By Proposition 4.1 $a < b$ implies $b \not{\preceq}_{\text{WL}} A a$. Because the $\preceq_{\text{WL}} A$ relation is total (Theorem 4.2) $a \preceq_{\text{WL}} A b$ holds.

Note, that the statement of the theorem without the second part of the premise is also correct. In contrast to the Pareto dominance relation it does not hold in general that $a \preceq_{\text{WL}} b$ implies $a \preceq b$ or vice versa. The definition of the WL relation implies that the order created on $A$ by the WL relation is equal to the lexicographical order induced by the pairs $(B^\succ(a,A), -W^\succ(a,A))$.

We extend relation $\text{WL}$ in the following such that for sets $A, B, C, D \subseteq X$

\[ A \preceq_{\text{WL}} C, D \Leftrightarrow B^\succ(A, C) > B^\succ(B, D) \land (B^\succ(A, C) = B^\succ(B, D) \land W^\succ(A, C) \leq W^\succ(B, D)) \]  

(5)

If $C = D$ we simply write $A \preceq_{\text{WL}} C B$.

Assume that $A$ is a set of solutions and that $B$ is the subset of size $k$ from $A$ that is best ranked with respect to the WL relation. Then a desirable property would be that $B$ ranks best (when compared analogously to other subsets of size $k$) with respect to its complement in $A$, i.e. $A - B$. Formally, we say property $(\ast)$ holds, if for all subsets $B, C \subseteq A$ with $|B| = |C|$: $B \preceq_{\text{WL}} A C \Rightarrow B \preceq_{\text{WL}} A - B A - C C$.

To see why property $(\ast)$ is desirable consider an example from sports. Let $B$ be a subset of the set $A$ of all teams in a league and let $S(B, A)$ be the the sum of the total score of all teams in $B$ for all games against the teams in $A$. Let $\text{games}$ be a relation that compares sets of teams with respect to the sum of total scores, i.e. $B \preceq_{\text{games}} A C$ if $S(B, A) > S(C, A)$. If $(\ast)$ holds for relation $\text{games}$ the teams in $B$ cannot profit from (illegal) agreements about the outcome of the internal games in $B$, i.e. the games between teams of $B$. To see this, assume that for a set of teams $C \subseteq A$ with $|C| = |B|$ it holds that $B \preceq_{\text{games}} A C$. Property $(\ast)$ implies $B \preceq_{\text{games}} A - B A - C C$. Thus $S(B, A - B) > S(C, A - C)$. Therefore, team $B \preceq_{\text{games}} A C$ because team $B$ was better than $C$ in the games against the respective other teams of $A$ but not because of the outcome of the internal games with $B$ respectively $C$.

The following example shows that property $(\ast)$ does not hold in general for the WL relation. Consider a directed graph $G = (V, E)$ with $V = \{v_1, \ldots, v_{11}\}$. Let $A = \{v_1, \ldots, v_5\}$, $B = \{v_6, \ldots, v_{10}\}$, and $C = \{v_{11}\}$. For every node $v \in A$ edge $(v_{11}, v) \in E$. All other edges in $E$ are
only between nodes of \( B \) such that every node in \( B \) has two ingoing edges from other nodes in \( B \) (details see Figure 1). As shown in Lemma 3.1 there exists an objective function \( f \) such that \( G \) is the favour graph of \( X = V \). Then \( B^*(b, X) = 2 \) for each node \( b \in B \), \( B^*(a, X) = 1 \) for each node \( a \in A \), and \( B^*(v_{11}, X) = 0 \). Thus, set \( B \) contains the five highest ranked nodes with respect to the WL relation. Set \( A \) contains nodes which are lower ranked than every node in \( B \) but higher ranked than \( v_{11} \) and node \( v_{11} \) is lower ranked than any node in \( A \cup B \). Therefore, \( B \sim_{WL} A \). But \( A \sim_{WL} \forall x \in A \sim_{WL} B \) since \( B^*(b, X - B) = 0 \) for each \( b \in B \) and \( B^*(a, X - A) = 1 \) for each \( a \in A \).

The relation that is introduced in the next section satisfies property (\( + \)).

4.2. Relation Points - Scoring With Ties

A potential problem of relation WL is that it ignores ties. Therefore it is proposed here to use a generalization of the scoring scheme which is also used in many sport leagues and tournaments where a team gets 2 (or 3) points for each won game and 1 point for each tied game. Inspired by this scheme we introduce the point score of a solution \( a \in X \) with respect to a set of solutions \( A \subseteq X \).

\[
S(a, A) = w \cdot B^*(a, A) + 1 \cdot E^*(a, A)
\]  
(6)

where \( w \geq 1 \) is a constant specifying how many points are attributed to a won comparison. For sets \( A, B \subseteq X \) define

\[
S(B, A) = \sum_{a \in B} S(a, A)
\]  
(7)

Based on this performance measure the Points relation is defined. For \( a, b \in X \) and \( A \subset X \) define

\[
a \sim_{\text{Pt}}^A b \iff S(a, A) \geq S(b, A)
\]  
(8)

The Points relation can be extended naturally (in the sense that \( b \sim_{\text{Pt}}^A c \iff \{b\} \sim_{\text{Pt}}^A \{c\} \)) to the set of subsets of \( X \) such that for sets \( A, B, C \subseteq X \)

\[
B \sim_{\text{Pt}}^A C \iff S(B, A) \geq S(C, A)
\]  
(9)

If it is clear from the context which set is the reference set \( A \), we omit the reference set in the notation.

**Theorem 4.4.** For \( A \subseteq X \) the Points relation \( \sim_{\text{Pt}}^A \) is a total preorder on \( X \).

**Proof.** Transitivity and totality are a consequence from \( \geq \) being transitive and total. Reflexivity is a consequence of totality.

The theorem ensures that it is possible to use the Points relation for sorting a set of solutions with a generic sorting algorithm, e.g. mergesort. Note, that the Points relation is not a partial order since antisymmetry does not hold, i.e. \( a \sim_{\text{Pt}} b \) and \( b \sim_{\text{Pt}} a \) does not imply \( a = b \).

**Example 4.2.** Consider Example 4.1. Since there are no ties it follows that \( a \sim_{\text{Pt}} b \sim_{\text{Pt}} c \sim_{\text{Pt}} a \). Since, \( a, b, c \) are pairwise different \( \sim_{\text{Pt}} \) is not antisymmetric.

**Proposition 4.2.** Let \( A \subseteq X \) and \( a, b \in A \) then \( a \prec b \iff b \not\sim_{\text{Pt}}^A a \).

**Proof.** Assume \( a \prec b \). By Theorem 4.1 we have \( B^*(a, A) > B^*(b, A) \). Any \( x \in A \) which contributes to \( E^*(b, A) \) contributes one point to \( S(b, A) \) and it holds that \( b \not\prec_f x \wedge x \not\prec_f b \). Since \( x \not\prec_f b \) we have by Proposition 3.1 that \( x \not\prec_f a \). Hence, \( x \not\prec_f a \) does not hold. Thus, \( x \) contributes at least one point to \( S(a, A) \) (one point in case of \( a \not\prec_f x \) and \( w \) points in case of \( a \prec_f x \)). Altogether, it follows that \( S(a, A) > S(b, A) \) and the theorem holds.
Theorem 4.5. The Points relation \( \prec_{p}^{A} \) is a refinement of \( < \) on \( A \), i.e. for \( a, b \in A \):
\[
a < b \land b \not< a \Rightarrow a \prec_{p}^{A} b \land b \not\prec_{p}^{A} a.
\]

Proof. The premise \( a < b \) implies by Proposition 4.2 \( b \not\prec_{p}^{A} a \). Since the Points relation is total (Theorem 4.4) \( a \prec_{p}^{A} b \) follows.

Note, that the statement of the theorem without the second part of the premise is also correct. Note also that, for \( w > 1 \) the Points relation might prefer dominated solutions over non dominated ones. As an example consider a set \( X = \{a, b, c\} \cup D \), with \( |D| \geq 2 \), \( a < c \) and \( \forall d \in D : a < d \land c < d \) (and all other pairs are incomparable). Hence, \( S(b, X) = |X| = |D| + 3 \), \( S(a, X) = 2 + w(|D| + 1) \), \( S(c, X) = 2 + w(|D|) \). Thus, \( c \) is ranked better than \( b \) if \( w > 1 \).

In sports \( w = 2 \) is often used for games where ties occur only rarely or never. However, in cases were ties are very likely to occur often a higher \( w \)-value is used. An example is the soccer league where \( w = 3 \) should encourage the teams to avoid ties and play for a win. In Volleyball there is nowadays a distinction between clearly won games that yield 3 points and close games which give 2 points. An analogous extension of relation Points is left as future work.

For the Points relation with \( w = 2 \) an interesting property holds as shown in the following. For \( A \subseteq X \) it holds that \( S(A, A) = |A|^2 + (w - 2) \cdot 2^{|A|} \). Thus, \( w = 2 \) is the only value for \( w \) where the score \( S(A, A) \) of a set \( A \) depends in general only on the size of the set and not on parameter \( w \). In the following we explain why this property is advantageous when the Points relation is used for the selection of a subset of \( k \geq 1 \) good solutions from a set of non dominated solutions. For this we extend the Points relation such that for sets \( A, B, C, D \subseteq X \) we define \( A \prec_{p}^{C,D} B \) iff \( S(A, C) \leq S(B, D) \). Assume that \( A \) is a set of non dominated solutions and \( B, C \subseteq A \) with \( |B| = |C| = k \) and \( S(B, A) \geq S(C, A) \). Then, it holds that \( S(B, A - B) = S(B, A) - S(B, B) = S(B, A) - k^2 \geq S(C, A) - k^2 = S(C, A) - S(C, C) = S(C, A - C) \). This proves that for \( w = 2 \) and sets \( A, B, C \subseteq X \) with \( |B| = |C| \) property (*)& does not hold in general. It can be shown that \( S(B, A) \) might exist that has a better score than \( B \) with respect to its corresponding complement, i.e. \( S(C, A - C) > S(B, A - B) \).

In the following we give two examples which show that \( S(B, A) \) might be the comparison of elements of \( B \) with each other leads to a high (small) number of wins with respect to \( < \) for \( w > 2 \) (respectively \( 1 \leq w < 2 \)). Then a subset \( C \subseteq A \) with \( |C| = k \) and \( S(C, A) < S(B, A) \) might exist that has a better score than \( B \) with respect to its corresponding complement, i.e. \( S(C, A - C) > S(B, A - B) \).

In the following we give two examples which show that \( S(B, A) \) might be the comparison of elements of \( B \) with each other leads to a high (small) number of wins with respect to \( < \) for \( w > 2 \) (respectively \( 1 \leq w < 2 \)). Then a subset \( C \subseteq A \) with \( |C| = k \) and \( S(C, A) < S(B, A) \) might exist that has a better score than \( B \) with respect to its corresponding complement, i.e. \( S(C, A - C) > S(B, A - B) \).
A summary of the properties of the ranking relations that have been studied in this paper is given in Table 2. Note, that by requiring $<$ instead of $\leq$ in the definition of the WL relation (last inequation of Equation (4)) one would obtain the corresponding strict weak ordering relation, i.e. a relation which is irreflexive, asymmetric, transitive, and where incomparability is transitive. Analogously, this holds for the Points relation when requiring $>$ instead of $\geq$ in Equation (8).

<table>
<thead>
<tr>
<th>Relation</th>
<th>Refinement of $&lt;$</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Favour</td>
<td>yes (Theorem 3.1)</td>
<td>irreflexive, asymmetric, not transitive, not total (Drechsler et al., 2001)</td>
</tr>
<tr>
<td>WL</td>
<td>yes (Theorem 4.3)</td>
<td>total preorder (Theorem 4.2)</td>
</tr>
<tr>
<td>Points</td>
<td>yes (Theorem 4.5)</td>
<td>total preorder (Theorem 4.4)</td>
</tr>
<tr>
<td>W</td>
<td>yes (Section 3.2)</td>
<td>total preorder (Section 3.2)</td>
</tr>
<tr>
<td>Win</td>
<td>yes (Section 3.2)</td>
<td>total preorder (Section 3.2)</td>
</tr>
</tbody>
</table>

5. The Metaheuristics

In this Section we describe the two metaheuristic algorithms that are used to test the new ranking relations. As test problem we use a multi objective scheduling problem with the aim to find a permutation of $n$ given jobs that is optimal for given set of cost functions (details are described in Section 6).

5.1. P-ACO

In contrast to the traditional ACO algorithms (for an overview see Dorigo and Stützle (2004)) the Population-based ACO (P-ACO) as proposed by Guntsch and Middendorf (2002) uses a population $P$ of $p$ solutions that is transferred from one iteration to the next (see Algorithm 1). The population is then employed to compute the pheromone information that is used by the ants to compute a set $L$ of $l$ new solutions. The pheromone information for the scheduling problem is stored in a pheromone matrix $[\tau_{ij}]$, $i, j \in [1:n]$ where $\tau_{ij}$ is the pheromone value for placing job $j$ at place $i$ in the schedule. Each pheromone value $\tau_{ij}$ is defined by $\tau_{ij} = \tau_{\text{init}} + \Delta \cdot k$ using $\Delta = (\tau_{\text{max}} - \tau_{\text{init}})/p$, where $\tau_{\text{init}}$ and $\tau_{\text{max}}$ are parameters of the algorithm giving the minimum and maximum pheromone values and $k$ is the number of permutations in the current population that have job $j$ at position $i$. As in traditional ACO algorithms an ant constructs a solution iteratively such that the probability $p_{ij}$ to select job $j$ for position $i$ in the permutation is $p_{ij} = \tau_{ij}/\sum_{k \in S} \tau_{ik}$ where $S$ is the set of jobs that are still selectable, i.e. the set of jobs that have not already been placed on positions 1 to $i-1$ in the permutation. This way of using the pheromone information is particularly suitable for scheduling problems (see also Merkle and Middendorf (2014)).

In the original P-ACO algorithm in every iteration the best solution from the set of new solutions $L$ is added to $P$ and the oldest solution is removed from $P$. Alternative strategies for the removal of solutions from $P$ have also been proposed in Guntsch and Middendorf (2002), e.g. to remove the worst solution. In this paper we use a different population update method that is based on the ranking relations: the solutions in $P \cup L$ are sorted with one of the ranking methods and the best $p$ solution form the new population $P$ for the next iteration. A stable sorting algorithm enforces that solutions of the population are not replaced by solutions that are not better.
Algorithm 1: Multi objective P-ACO with ranking

1. \( P \leftarrow \{\} \);
2. Initialize pheromone matrix \( [\tau_{ij}] \leftarrow \tau_{init} \);
3. \( \text{repeat} \)
4. \( L \leftarrow \{\} \);
5. \( \text{foreach of the ants} \)
6. Construct a solution \( S \) using pheromone matrix \( [\tau_{ij}] \);
7. \( L \leftarrow L \cup \{S\} \);
8. Sort the solutions in \( P \cup L \) with a ranking method;
9. Let \( P \) be the set of \( p \) best solutions from \( L \cup P \);
10. Compute a pheromone matrix \( [\tau_{ij}] \) from \( P \);
11. \( \text{until} \) \( t \) iterations done;
12. \( \text{return} \) non dominated solutions in \( P \cup L \)

5.2. GA

We use a standard genetic algorithm (GA) to investigate the influence of the different ranking methods when the algorithm is applied to the multi objective scheduling problem (for an introduction to genetic algorithms see, e.g. Mitchell (1996)). The GA uses a population \( P \) of \( p \) solutions that are transferred from one iteration to the next. In every iteration a set \( L \) of \( l \) new solutions is computed. The new solutions are obtained by two types of operations from solutions in \( P \): i) crossover of two solutions (called parents) from \( P \) and ii) mutation of a solution in \( P \). As crossover operation a two point crossover is used, where the interval between two random points of a permutation is taken from one parent and the other positions are filled by the remaining jobs in the order of occurrence in the other parent. As mutation operator a switch operation is used to exchange the order of two neighbored jobs in a permutation.

To create the \( l \) new solutions from the parent population each possible pair of solutions is taken to create one new solution by crossover. Moreover, each solution from \( P \) is taken to create one new solution by mutation. Hence, altogether \( l = p \cdot (p - 1) + p = p^2 \) new solutions are created per iteration. The \( p \) old solutions from \( P \) together with the \( p^2 \) new solutions are ranked with one of the ranking methods. Then the best \( p \) solutions are selected to form the new population \( P \) for the next iteration. A stable sorting algorithm is applied to ensure that solutions in the population are not removed from the population in favor of an equally ranked solution, that was generated in the respective iteration.

5.3. Algorithm Variants

The different P-ACO and GA algorithms are denoted by X–P-ACO, respectively X–GA, where \( X \in \{Pt, WL, W, Win\} \) is the respective ranking method that is used in the algorithm. When the type of algorithm is clear from the context \( X \) is used as abbreviation of X–P-ACO or X–GA.

The P-ACO with the new ranking methods was also compared to two multi objective P-ACO algorithms from the literature: i) the Crowding P-ACO (Cr–P-ACO) introduced in Angus (2007) and ii) a P-ACO which is called Standard P-ACO (Std–P-ACO) here and is motivated by Guntsch and Middendorf (2003). In the Crowding P-ACO the amount of updated pheromone is inversely proportional to the dominance depth of the corresponding solution (Deb et al., 2000).
The population update is done by a crowding scheme. In each iteration for each newly generated solution a random subset of the population is chosen. If the new solution dominates the solution of this subset, that has the highest similarity with the new solution, the new solution replaces that solution. Otherwise the new solution is discarded. Similarity is measured in the solution space as the number of common adjacent jobs. Note, that this is different from the crowding measure that was used in Deb et al. (2000), where similarity was analyzed with respect to the objective space. The Standard P-ACO computes the non dominated front from the solution in \( P \cup L \) in every iteration. The new population for the next iteration is determined by selecting a randomly chosen solution \( s \) from the non dominated front and then choosing the \( p - 1 \) solutions closest to \( s \) in the objective space. Note that the \( Cr- \) and \( Std- \) P-ACO make no use of the new ranking methods, but are based solely on the Pareto dominance relation.

6. Experiments

In this section we present the scheduling problem used for the empirical evaluation of the metaheuristics presented in Section 5 using the ranking methods described in Sections 3 and 4. Furthermore the parameters used in the algorithms are described.

6.1. Multi Objective Flow Shop Scheduling

For the empirical evaluation we consider a multi objective version of the Flow Shop Scheduling Problem that was shown to be NP-hard (Garey et al., 1976). The problem is to find a processing sequence, i.e. a permutation, of \( n \) given jobs that have to be processed by \( m \) machines in a given order. Every job has to be processed by every machine for a time that is specific for each combination of job and machine. Each job starts at the first machine. When a job is finished at the \( i \)-th machine it is passed to the \((i + 1)\)-th machine. The last machine of each job is the \( m \)-th machine. A machine can only process one job at a time and a job can only be processed by one machine at a time. It is assumed here that all jobs are available at time 0. The considered multi objective version of the Flow Shop Scheduling Problem has four objectives that are to be minimized: (i) \( \text{Total Time } T_{total} \), i.e. the completion time of the last job on the last machine, (ii) \( \text{Flow Time } T_{flow} \), i.e. the sum of the processing and idle times of the jobs, (iii) \( \text{Total Idle Time of Jobs } T_{i,j} \), i.e. the sum of all waiting times of all jobs after the job was finished on a machine and before it is processed on the next machine, (iv) \( \text{Total Idle Time of Machines } T_{i,m} \), i.e. the sum of all idle times of all machines where a machine has an idle time when it is unoccupied between the processing of two jobs.

The P-ACO and the GA algorithm have each been applied to 16 test instances shown in Table 3. From the available data sets the largest instances were chosen plus car1 because it is possible to compute the Pareto front for this small instance. In order to characterize Multi Objective Flow Shop Scheduling in more detail we present the correlation between the objectives for car1 in the next subsection.

6.2. Correlation of the Objectives

As typical for many multi objective optimization problems the different objectives are not independent from each other. Table 4 shows the correlations between the objectives for the

---

\( ^1 \)Taken from the OR Library: people.brunel.ac.uk/~mastjjb/jeb/info.html.
Table 3: Used data sets and their properties.

<table>
<thead>
<tr>
<th>Name</th>
<th>From</th>
<th>n</th>
<th>m</th>
</tr>
</thead>
<tbody>
<tr>
<td>car1</td>
<td>Carlier (1978)</td>
<td>11</td>
<td>5</td>
</tr>
<tr>
<td>car8</td>
<td>Carlier (1978)</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>hel1</td>
<td>Heller (1960)</td>
<td>100</td>
<td>10</td>
</tr>
<tr>
<td>reC37,39,41</td>
<td>Reeves (1995)</td>
<td>75</td>
<td>20</td>
</tr>
<tr>
<td>ta101-ta105</td>
<td>Taillard (1993)</td>
<td>200</td>
<td>20</td>
</tr>
<tr>
<td>ta111-ta115</td>
<td>Taillard (1993)</td>
<td>500</td>
<td>20</td>
</tr>
</tbody>
</table>

Table 4: Pearson correlation (all p-values < 0.01) for the Pareto set and a random set of solutions (502 solutions each) for the car1 instance. Total Time ($T_{total}$), Flow Time ($T_{flow}$), Total Idle Time Jobs ($T_{i,j}$), and Total Idle Time Machines ($T_{i,m}$).

<table>
<thead>
<tr>
<th></th>
<th>$T_{flow}$</th>
<th>$T_{i,m.}$</th>
<th>$T_{i,j}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pareto</td>
<td>0.512</td>
<td>-0.404</td>
<td>0.179</td>
</tr>
<tr>
<td>random</td>
<td>0.675</td>
<td>-0.128</td>
<td>0.676</td>
</tr>
<tr>
<td>Pareto</td>
<td>-0.813</td>
<td>0.382</td>
<td>0.289</td>
</tr>
<tr>
<td>random</td>
<td>-0.325</td>
<td>0.771</td>
<td></td>
</tr>
</tbody>
</table>

Pareto set and for a random set of solutions for the car1 problem instance. For the Pareto set the strongest absolute correlation value is the (negative) correlation between the total idle times of the machines $T_{i,m}$ and of the jobs $T_{i,j}$. The second strongest absolute correlation value is the also negative correlation between the flow time $T_{flow}$ and $T_{i,m}$. The two strongest positive correlations are between $T_{flow}$ and $T_{i,m}$ and between $T_{total}$ and $T_{flow}$.

The value of all correlations for the random set of solutions is larger than for the Pareto set. However, the signs of the corresponding correlations are identical with the single exception of the correlation between $T_{total}$ and $T_{i,m}$ which is negative in the Pareto set, but positive in the random set. In the random set the strongest correlations are the two positive correlations between $T_{flow}$ and $T_{i,j}$ and between $T_{total}$ and $T_{flow}$. Both of these correlations are stronger than in the Pareto set. The two negative correlations between $T_{i,m}$ and $T_{i,j}$ and between $T_{flow}$ and $T_{i,m}$ are much weaker for the random set of solutions than for the Pareto set.

As typical for many application problems these more or less striking differences in the correlations between objectives are a result of the special characteristics of the problem. They show also that solutions in the Pareto set have different characteristics than random solutions. The mixture of strong and weak as well as positive and negative correlations highlights the difficulty of the provided multi objective optimization problem.

6.3. Parameters

For the sake of comparability no heuristic is used in any of the algorithms. In Pt–P-ACO and Pt–GA relation Pt is used with $w = 2$. For W–P-ACO and W–GA randomly with uniform probability chosen weights $w_i \geq 0$ for $i \in [1 : d]$ and $\sum_{i=1}^{d} w_i = 1$ are used. The weights are chosen in every iteration anew and are averaged with those of the previous iteration. By taking the averages the weights change only gradually.
For the GA and the P-ACO every experiment consists of 25 runs for each algorithm. Each algorithm was run for $t = 50,000$ iterations. The P-ACO and the GA population size is $p = 5$. In each iteration $l = 25$ new solutions are created. The proposed ranking methods are applied to choose the five best solutions out of the $p$ solutions in the population and the $l$ new solutions. To improve the comparability of the Crowding P-ACO the population size and the number of ants is set to 25, i.e. $l = p = 25$, and the size of the random subsets is set to 5. Populations of this size have been shown to perform well (Gunstsch and Middendorf, 2002, 2003; Oliveira et al., 2011). For the pheromone update of the P-ACO parameter values $\tau_{\text{init}} = 1/n$ and $\tau_{\text{max}} = 25$ were used as suggested by the results of Gunstsch and Middendorf (2003). For the GA the mutation operator had a mutation probability of $1/(n - 1)$ for each position. Table 5 gives an overview over the parameters and their values.

### Table 5: Parameters used in experiments.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Values for Experiments to Infer Solution Quality and Convergence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iterations</td>
<td>$50,000$ $100,000$</td>
</tr>
<tr>
<td>Runs</td>
<td>$25$ $10$</td>
</tr>
<tr>
<td>Population</td>
<td>$5$ $5$</td>
</tr>
<tr>
<td>Number of Ants</td>
<td>$25$ $25$</td>
</tr>
<tr>
<td>$\tau_{\text{init}}$</td>
<td>$1/n$ $1/n$</td>
</tr>
<tr>
<td>$\tau_{\text{max}}$</td>
<td>$25$ $25$</td>
</tr>
<tr>
<td>Mutation Probability</td>
<td>$1/(n - 1)$ $1/(n - 1)$</td>
</tr>
<tr>
<td># Solution Evaluations</td>
<td>$1,250,000$ $2,500,000$</td>
</tr>
</tbody>
</table>

7. Results

The experimental comparison of algorithms that solve multi objective optimization problems is not trivial. One reason is that, typically, there is not a single best solution for each algorithm but a set of different (non dominated) solutions. Here we concentrate our analysis on the distribution of the solutions found by the different algorithms and how much they dominate the solutions of the respective other algorithms. Additionally we infer which algorithm scored best concerning each of the four objectives separately. As the true Pareto front is not known in all but one of the instances (namely car1) it is unclear how many of the found solutions are actually Pareto optimal.

7.1. Analysis of the Non Dominated Front

Table 6 shows the average size of the non dominated front obtained by the P-ACO and GA algorithms for the different problem instances. The non dominated front at the end of a run is computed from the set $L$ of the $l = 25$ new solutions of the final iteration and the $p = 5$ solutions in the population $P$. For algorithms Pt–P-ACO, WL–P-ACO, and Win–P-ACO the non dominated front is on average smaller than 5 (and for each instance smaller than 6). Most - if not all - solutions generated by the ants in the final iteration are dominated by the solutions that are in the population. This may indicate the high quality of the solutions in the population.
Table 6: Average size of the non dominated fronts.

<table>
<thead>
<tr>
<th></th>
<th>P-ACO</th>
<th>GA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>{Pt WL W Win Cr Std}</td>
<td>{Pt WL W Win}</td>
</tr>
<tr>
<td>car1</td>
<td>4.8</td>
<td>4.7</td>
</tr>
<tr>
<td>car2</td>
<td>2.8</td>
<td>4.4</td>
</tr>
<tr>
<td>car3</td>
<td>4.9</td>
<td>5.4</td>
</tr>
<tr>
<td>car4</td>
<td>4.8</td>
<td>4.6</td>
</tr>
<tr>
<td>car5</td>
<td>5.1</td>
<td>5.0</td>
</tr>
<tr>
<td>car6</td>
<td>4.7</td>
<td>5.0</td>
</tr>
<tr>
<td>car7</td>
<td>4.9</td>
<td>5.4</td>
</tr>
<tr>
<td>car8</td>
<td>2.8</td>
<td>4.4</td>
</tr>
<tr>
<td>car9</td>
<td>4.9</td>
<td>5.5</td>
</tr>
<tr>
<td>car10</td>
<td>4.8</td>
<td>5.7</td>
</tr>
<tr>
<td>car11</td>
<td>5.0</td>
<td>5.0</td>
</tr>
<tr>
<td>car12</td>
<td>4.8</td>
<td>5.0</td>
</tr>
<tr>
<td>car13</td>
<td>4.9</td>
<td>5.4</td>
</tr>
<tr>
<td>car14</td>
<td>2.8</td>
<td>4.4</td>
</tr>
<tr>
<td>car15</td>
<td>4.9</td>
<td>5.0</td>
</tr>
<tr>
<td></td>
<td>4.7</td>
<td>4.5</td>
</tr>
</tbody>
</table>

In contrast, the non dominated fronts of Cr–P-ACO and the Standard P-ACO contain on average more than 6 solutions. Hence, in these cases, some of the new solutions from the last iteration are not dominated by the solutions in the population.

The non dominated fronts of the GAs are far smaller. GAs need a set of heterogeneous solutions to construct new solutions by crossover that are different enough from the parental generation. The populations of five solutions that are generated by the algorithms, however, seem to be too small to provide sufficient heterogeneity. The newly generated solutions differ only slightly from each other, resulting quite likely in a small non dominated front.

To compare the impact of the ranking relations \(\succ_{Pt}\) and \(\succ_{WL}\), the combined non dominated front of the union of the solutions of the Pt–P-ACO (WL–P-ACO) with the competitor algorithms (i.e.

Table 7: Percentage of solutions in the combined non dominated front (upper row): with Pt (left side) or with WL (right side); absolute average size of the front with standard deviation (lower row); values are averages over all problem instances. The gray values in each group are the values that are not dominated significantly by any other value in the group (Mann-Whitney significance test, \(p\text{-Value} < 0.01\)).

<table>
<thead>
<tr>
<th></th>
<th>Combined front of</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>{Pt W Win Cr Std}</td>
</tr>
<tr>
<td>P-ACO</td>
<td>39.6 26.1 33.5 1.3 5.0</td>
</tr>
<tr>
<td></td>
<td>72.1±17.0</td>
</tr>
<tr>
<td>GA</td>
<td>35.3 34.2 41.1 - -</td>
</tr>
<tr>
<td></td>
<td>31.3±14.4</td>
</tr>
</tbody>
</table>
Table 8: Fraction of solutions that rank in the top 10% of all solutions from the respective class of algorithms (P-ACO respectively GA) concerning one of the criteria with standard deviation; Total Time ($T_{total}$), Flow Time ($T_{flow}$), Total Idle Time Jobs ($T_{i,j}$), Total Idle Time Machines ($T_{i,m}$); values are averages over all problem instances; best values are shown in grey.

<table>
<thead>
<tr>
<th>Combined front of { Pt  W  Win  Cr  Std }</th>
<th>{ WL  W  Win  Cr  Std }</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PACO</strong></td>
<td></td>
</tr>
<tr>
<td>$T_{total}$ 58.4</td>
<td>6.1</td>
</tr>
<tr>
<td>$T_{flow}$ 42.5</td>
<td>25.1</td>
</tr>
<tr>
<td>$T_{i,m}$ 53.3</td>
<td>3.7</td>
</tr>
<tr>
<td>$T_{i,j}$ 5.0</td>
<td>87.3</td>
</tr>
<tr>
<td><strong>GA</strong></td>
<td></td>
</tr>
<tr>
<td>$T_{total}$ 40.6</td>
<td>5.3</td>
</tr>
<tr>
<td>$T_{flow}$ 30.1</td>
<td>39.8</td>
</tr>
<tr>
<td>$T_{i,m}$ 35.1</td>
<td>1.0</td>
</tr>
<tr>
<td>$T_{i,j}$ 2.1</td>
<td>92.9</td>
</tr>
</tbody>
</table>

W–P-ACO, Cr–P-ACO, Win–P-ACO, and Std–P-ACO) has been computed. The combined non dominated front is obtained by merging all solutions of the final iterations of all runs of the considered algorithms. The non dominated front of this set of solutions is denoted as the combined non dominated front or just combined front of the set of considered algorithms. Analogously this has been done for the GA algorithms. The fraction of solutions of each algorithm in the combined non dominated front of its class of algorithms (P-ACO respectively GA) is shown in Table 7. The table also shows the absolute size of the combined pareto front. Note, that some solutions in the combined non dominated front were detected by several of the algorithms, i.e. the sum of percentages within each class can be larger than 100%.

The table shows for the P-ACOs that Pt–P-ACO and WL–P-ACO obtained the largest fraction. Win–P-ACO obtained a slightly smaller fraction. However, the difference between Pt–P-ACO and Win–P-ACO (respectively WL–P-ACO and Win–P-ACO) is not statistically significant (with p-values $< 0.01$). The solutions found by Cr–P-ACO, and Std–P-ACO are mostly dominated by solutions of the other algorithms. Both algorithms detected on average less than 5% of the solutions in the combined non dominated front. The difference between their respective values are statistically insignificant. For the GAs each of the three algorithms Pt–GA, W–GA, and Win–GA (respectively, WL–GA, W–GA, and Win–GA) obtained a large part of the front. The differences between the three algorithms are statistically not significant (all p-values $> 0.01$).

7.2. Objective Oriented Analysis

In the following we consider the quality of the found solutions with respect to the different objectives. Table 8 lists the fraction of solutions that the algorithms found within the best 10% solutions of all compared algorithms with respect to each single objective.

For the P-ACOs the algorithms Pt–P-ACO and WL–P-ACO rank first in the three criteria Total Time, Flow Time, and Total Idle Time Machines. Win-P-ACO ranks second for these three criteria. For the Total Idle Time Jobs criterion algorithm W–P-ACO ranks first with a large difference to all other algorithms. However, W–P-ACO finds only few good solutions with respect to Total Time and Total Idle Time Machines. Cr–P-ACO and Std–P-ACO fail to produce competitive solutions.

The results for the GA algorithms show some similarities. Namely, that for the two criteria Total Time and Total Idle Time Machines algorithms Pt–GA and Win-GA (respectively WL–
P-ACO

GA

Figure 2: Histogram of the number of $B^<(a,b)$ values of all solutions $a$ in the combined non dominated fronts of one algorithm compared with all the solutions $b$ of all respective other algorithms; for the group of algorithms with Pt–P-ACO (left) and the group of algorithms with WL–P-ACO (right).

P-ACO and Win-GA) are the two best algorithms. However, different from the P-ACO results, Win–GA ranks mostly first. Similar as for the P-ACO results, algorithms Pt-GA, W-GA, and Win-GA (respectively WL–GA, W–GA, and Win–GA) are good for Flow Time. Also similar as for the P-ACO results, algorithm W-GA is clearly the best for Total Idle Time Jobs.

We consider the ability of a multi objective algorithm to detect solutions with good values in all or most objectives as a desirable property. Therefore we analyze the solutions in the combined non dominated front with the previously introduced measure $B^<(a,b)$ (see Figure 2). Recall, that with four criteria the values range from 0 to 3 as no solution can be worse in all criteria compared to any other solution of the non dominated front. The solutions of Pt–P-ACO and WL–P-ACO clearly obtain the largest number of comparisons where they are better in three objectives. This hold also for the number of comparisons where they are better in two or more objectives. The results obtained with the GA algorithms are similar. However, a difference is that Win-GA has a slightly larger number of comparisons where it is better in three objectives than Pt-GA.

7.3. Convergence Behavior

In the following we analyze the convergence behavior of the different P-ACO algorithms for one instance from each of the four classes of test problems: car1, hell, reC37, and iat111. For this we use the hypervolume indicator which measures the volume, that is dominated by the non dominated front and dominates a fixed reference point. Our reference point is a combination of the largest (worst) values of all objectives of all algorithms encountered during all experimental runs. The larger the absolute value of hypervolume indicator, the better is the set of solutions provided by the tested algorithm. The hypervolume indicator of the non dominated front of
solutions in the population plus the solutions generated by the ants in the respective iteration has been measured every ten iterations. The results are shown in Figure 3.

Figure 3: Average hypervolume indicator during 10 test runs with 100000 iterations measured every 10 iterations on four different instances: car1 (top, left), hel1 (top, right), reC37 (bottom, left), ta111 (bottom, right); reference point is the combination of all maximal values for each objective measured in the experimental runs; green (upper) and red (lower) horizontal lines for car1 give the average hypervolume indicator for 1000 samples with 5 respectively 30 randomly chosen solutions from the true Pareto front.

The figure shows that with the exception of Cr– and Std–P-ACO all algorithms converge during the first 75,000 iterations to a low level and show only minor improvement subsequently. In all four cases Pt–P-ACO obtains the best values. However, WL–P-ACO and Win-P-ACO obtain similarly good values and are second and third best. W-P-ACO is for all four instances better than Cr– and Std–P-ACO, but clearly worse than Pt–P-ACO, WL–P-ACO and Win-P-ACO. For car1 the solutions that are generated by Pt–P-ACO, WL–P-ACO, and Win-P-ACO dominate nearly as much hypervolume as 5 solutions of the Pareto front do on average, see Figure 3. Cr– and Std–P-ACO show an apparently random behavior with respect to the dominated hypervolume. Std–P-ACO has been stripped of its archive and therefore uses much less information than the original algorithm by Guntsch and Middendorf (2002) and thus fails to drive its ants towards better solutions. The sparse pheromone matrix, built solely by the few solutions in the population, appears to be insufficient for algorithms that rank solutions solely by Pareto dominance. The Cr–P-ACO is also an algorithm of that type, picking solutions for its population at random from the non dominated front of solutions.
8. Conclusions

Two new methods for ranking the solutions of multi objective optimization problems have been proposed. The new methods are called WL relation and Points relation, respectively and are based on the favour relation that was studied in Drechsler et al. (2001). The ranking methods can be used by iterative metaheuristics to select good solutions that are to be used in the next iteration from a set of solutions that have been found in the current iteration. It was shown theoretically that the new ranking methods have desirable properties that are useful for such an application. In particular, the ranking methods were related to the notion of Pareto dominance. It was shown, e.g. that both ranking methods form a total preorder and both are refinements of the Pareto dominance relation.

In an experimental study the ranking methods were used in a Population-based Ant Colony Optimization algorithm (P-ACO) and in a genetic algorithm (GA) to select the solutions that are allowed to enter the population. The P-ACO algorithms with the new ranking methods were compared with four other P-ACO algorithms, e.g. P-ACOs using other ranking methods and the Crowding P-ACO from Angus (2007). The GA algorithms with the new ranking methods were compared to two GAs that use other ranking methods. In particular, we studied a situation where the population size is kept small and the algorithms are applied to a four objective flowshop problem. We used no additional heuristic information. The results show that both metaheuristics profit from the new ranking methods and perform well on the flowshop problem.

For future work it is interesting to study the combination of several ranking methods within one algorithm. Our results indicate that a combination of one or both of Pt and WL ranking together with the Win ranking might perform very well. This could be good also for metaheuristics with a large population where the aspect of diversity is important.

Other values for the $w$ parameter of the WL ranking could also be of further interest. How the different ranking methods perform in other evolutionary multi objective algorithms and on other problems can be investigated.

Acknowledgment

We would like to thank the anonymous reviewers for their many, detailed, and very valuable comments and pointing out errors in a former draft. This helped to improve the paper. This work has been supported by the European Social Fund (ESF) and the Free State of Saxony within Nachwuchsforschergruppe „Schwarm-inspirierte Verfahren zur Optimierung, Selbstorganisation und Ressourceneffizienz“.

References


